Facial Keypoints Detection: Reloading Classical Techniques

Igor Borovikov

Center for Advanced Signal and Image Sciences (CASIS) at LLNL

19th Annual Workshop, May 13, 2015

Abstract

The presentation describes **work in progress** done using Kaggle facial keypoints database [1]. The goal is to achieve robust detection of facial keypoints like centers of eyes, mouth, nose tip, etc. on low-resolution real world imagery. The approach to the problem is shaped by the requirement of potential customer to exclude neural networks (as "too black boxy"). Instead we use well-known **classical Machine Learning** techniques such as **Eigenfaces**, **k Nearest Neighbors** combined with various computer vision techniques with intention to push those as far as possible in comparison to state of the art deep learning methods.

Effective detection and handling of outliers, dealing with training data idiosyncrasies proved yet again to be an important part of pipeline. In addition to simple data augmentation we use low level computer vision tricks like local adaptive histogram equalization (CLAHE) to improve the performance.

We start with an overview of attempted **direct detection (no explicit learning)** of facial keypoints with computer vision methods such as features localization with Haarcascades, local hierarchical matching of edge templates in various wavelet spaces and optical flow for image registration - mostly to outline their shortcomings in the context of facial keypoints detection.

Motivation, Data and Constraints

- **Facial keypoints detection can be first step** in many higher level tasks - from identifying subject, inferring emotional expressions to (partial) reconstruction in computer graphics.
- Obtaining **quality database of training data** could be challenging due to obvious technical constraints and legal aspects. For this work we mostly used Kaggle dataset [1] that also comes with tools of computing error on test data set.
- **Facial keypoint sets can vary** from a simple quadruple of eyes, mouth centers and nose tip (left) to more detailed ones that include corners of eyes, mouth and eyebrows (right):

Figure 1. Annotated training data images from two different subsets

- While artificial neural networks/deep learning appear to be the state of the art in the field, there remains interest in classical methods due to their "transparency" and easier ceiling analysis (**no "black boxes" requirement**).

Direct Detection

The first approach was based entirely on low level computer vision techniques and was applied to various free celebrities images pulled from the internet. They offered more room for experimentation due to higher resolution (but the image set was very sparse and shot mostly under controlled conditions).

For copyright and licensing considerations we are using author's image here.

- **Normalization** of images by scaling rotating to the target resolution in upright orientation. It is "chicken and egg" problem if the normalization relies on features localization.
- **Haarcascades for localizing features** (Python, OpenCV).

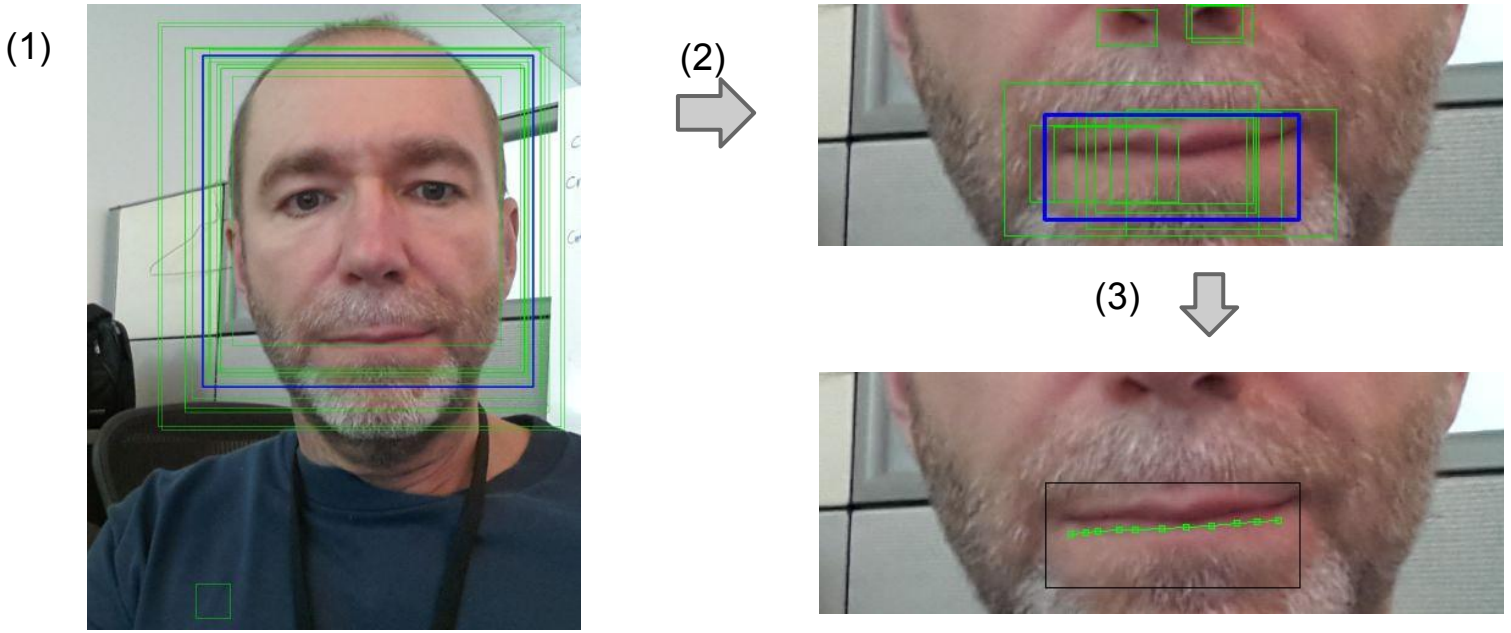


Figure 2. Hierarchical **feature localization** (1) and (2) followed by **contour fitting** (3).

Figure 2 comments:

(1) Outliers found even on "clean" images (small green false positive on the first image); hence the method has to be "robustified" by running it with different parameters, rejecting outliers and finding robust mean blue rectangle (multiple runs of Haarcascades are somewhat slow!).

(2) **Hierarchical** approach improves accuracy of subsequent stages by guessing better starting ROI.

(3) Sufficiently well localized features can be processed to extract finer details - contour fitting in this case.

Contour fitting was also done in hierarchical top-down manner using energy-minimization. The energy of a contour was computed on a non-uniform Sobel edges images computed on à trous wavelet transform [2] at level 3. Such pre-processing eliminates lighting and leaves only gross edge features on the image without loosing resolution required for fine contour matching.

Downsides:

- Heavy expensive preprocessing,
- **Easily fails on less "clean" images!**

Some of the failures could be addressed by simultaneous detection of features supporting each other, e.g. an angle of the line between eyes centers is about the same as mouth line angle:

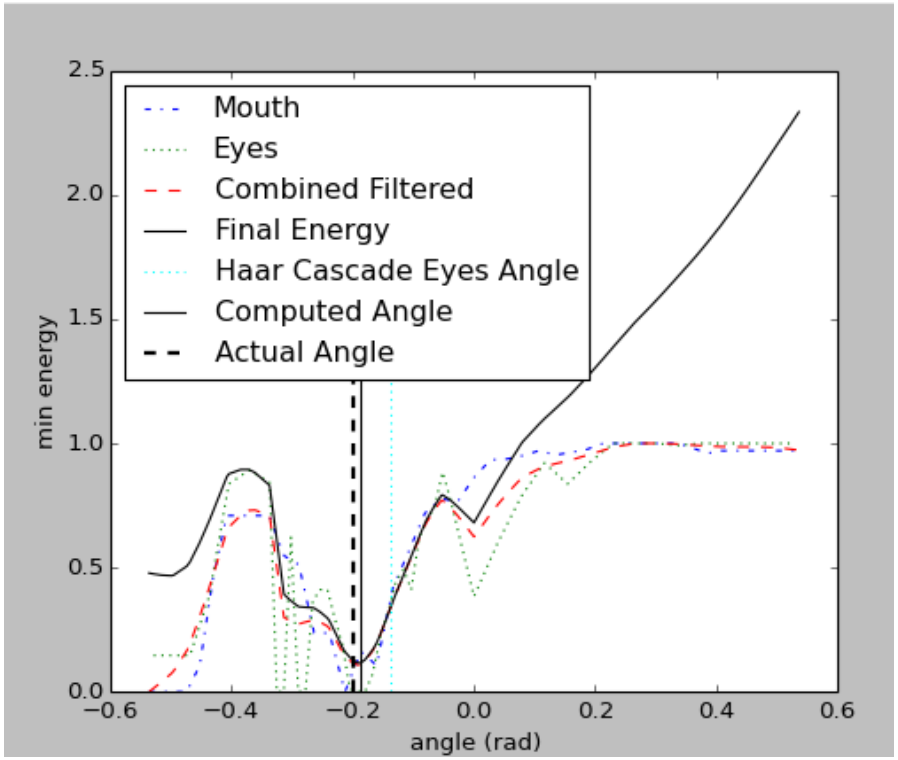
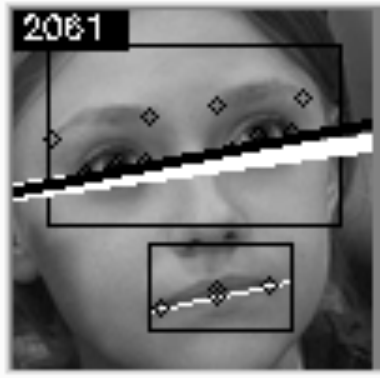


Figure 3. Below: training image with debug drawing for eyes/mouth combined line detected. Left: energy curves for detection components. The angle is computed using several features (locations of Haar eyes, mouth two points contour, eye dominant direction) with individual angles are combined as shown on the plot to compute single value.



While the combined technique is more robust, it leaves many holes unplugged. Direct approach discarded:

- lacks robustness, sensitive to data, heavy pre-processing;
- unmanageable number of parameters leading to problems with their optimization.

Eigenfaces and kNN

Classical approach with off-shelf open source tools.

Data pre-processing and augmentation:

- Local adaptive histogram equalization (CLAHE);
- Mirrored images added to the training database (doubling 7050 training images and removing asymmetry).
- Resampling 96x96 => 32x32
- PCA for 150 principal components (approx. 60% explained variance):

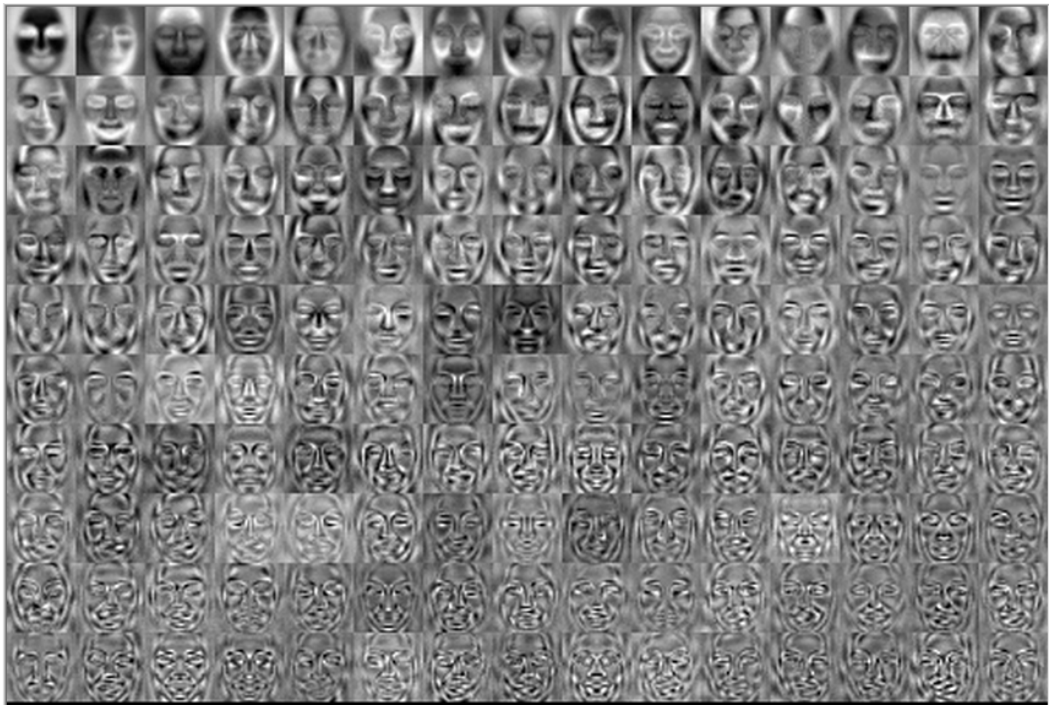


Figure 4. Principal vectors: first image is true first principal component, the rest are difference from the previous one (for better visualization).

kNN in Principal Components Basis:

Figure 5. Top left: **test** image.

It is followed by first 17 nearest neighbors from training data set. Target image was identically pre-processed
CLAHE=>
resample =>
Principal Components
to bring it to the same space.

The predicted landmarks computed as weighted mean from the nearest neighbor values.

Note that the image orientation of first neighbors matches the test image and they are close in appearance, as expected from PCA definition.

Performance Measure MSE (Mean Squared Error)

Training and Cross Validation Errors

"Naive" training error = 0: there is always a direct hit in the set of nearest neighbors. Cheap solution for trivial training error: "**leave one out**", i.e. discard first neighbor.

Better, still not perfect; remains issue of **mirrored direct hits**. Hence "cross validation" and "training error" used here interchangeably.

Test Error

Kaggle online tool - measuring error with test imagery. Still direct hits occur: test data comes from the same source(s).

First Iteration Results

	Training Error*	Test Error**
Mean of kNNs in PCA	13.0	12.0
Weighted Mean (see next page)	5.3	4.1
Weighted Mean over Data Subsets (see next page)	2.8	3.3

*), **) "Leave one out" removes all "direct hits" when computing **training error**, so it could be greater than **test error** where direct hits still can (luckily!) reduce total error.

Computing keypoints location using kNNs in principal component space as follows:

Simple mean *Pm* - prediction, *pi* - neighbors values for *i*-th keypoint:

$$P_m = \sum_{i=1,n} \frac{p_i}{n}$$

Weighted Mean for kNNs is computed as:

$$P_m = \sum_{i=1,n} \frac{w_i p_i}{W}$$

where:

$W = \sum_{i=1,n} w_i$ - normalization,

$w_i = \frac{\hat{d}}{(d_i+\epsilon)(1+i)}$ - *i*-th neighbor weight, ϵ - regularization for direct hits (zero min distance),

$\hat{d} = \min_{i=1,n} d_i$ - Euclidean distance for the first nearest neighbor.

Data subsets are taken into account by setting weight to zero for neighbor from an incompatible for current keypoint subclass.

Similarly, images from original and mirrored pair are used only once - using both leads to undesirable “symmetrization” of the prediction.

Training and Test Data Idiosyncrasies

- two subsets of training data with different definitions of landmarks for nose tip and bottom lip center: nearly but not entirely separable by number of provided keypoints.
- duplicate images or same person present many times with different facial expression (skewes PC space):

- intentionally (?) injected "noise", degraded quality, etc:

- various style images (grayscale, binary, sketches, paintings, sunglasses, objects, writings, etc):

- incorrect keypoints in training dataset;
- two faces on single image in test data.

Image Registration

Attempting to bridge gap from kNN to the target test image with image registration:

- Haarcascades face location shift-scale produced poor results due to limited precision of the face location and frequent failures to localize it at all.
- Optical Flow (Pyramidal Lucas Kanade optical flow with following "tracking" of sparse points): fails because optical flow assumptions are not holding in arbitrary pair of images, even with histogram equalization and heavy blurring; also keypoints are not necessarily "good features to track".
- Affine Registration: using three non-ambiguous key points (eyes centers, bottom lip center) minimize weighted energy:
  - higher weight on vicinity of keypoints,
  - penalizing for leaving image boundaries.

	Training	Test
Affine predictor	(2.8)*	3.1

*) See footnotes on pg.9

Issues:

- Absolute energy is not necessarily an indicator of the matching quality; Use mutual information?
- Combining with kNN weighted average is not straightforward (correlation energy has different scales).

Discussion and Future Work

- no significant parameters optimization done yet; the main parameters are resampling size, number of principal components to compute - higher values of cumulative proportion of variance explained (CPVE) by each principal component may help;
- adjusting weights for vector components for kNN basing on relative entropy (mutual information);
- adding affine transformations as data augmentation, in addition to mirrored images - this will likely speed up affine matching or will make it unnecessary;
- using salience of regions as weight in image registration; salience can be computed using the entire dataset;
- possible incorporation of Stasm framework (active shapes model) [4]
- refining PCA+kNNs approach with direct methods discussed in the first part;
- perform pattern match in appropriate image space.

References

[1] Kaggle Online Competition: Facial Keypoints Detection. https : //www.kaggle.com/c/facial-keypoints-detection.

[2] Shensa, M.J., 1992. Discrete Wavelet Transforms: Wedding the a trous and Mallat algorithms. IEEE Transactions on Signal Processing, 40, pp. 2,464-2,482.

[3] Facial Keypoints Detection, Y. Wang and Y. Song, Stanford University http://cs229.stanford.edu/proj2014/Yue%20Wang,Yang%20Song,Facial%20Keypoints%20Detection.pdf

[4] S. Milborrow, F. Nicolls Active Shape Models with SIFT Descriptors and MARS. VISAPP2014 (http://www.milbo.users.sonic.net/stasm/)

[5] Shijian Lu, Cheston Tan, Joo-Hwee Lim. Robust and Efficient Saliency Modeling from Image Co-Occurrence Histograms IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 36, No. 1, January 2014.